

REMARKS/ARGUMENTS

The claim amendments and arguments presented herein include the claim amendments and arguments Applicants discussed with the Examiners during the phone interview on October 24, 2006. Applicants submit that the arguments and amendments presented herein make the substance of the phone interview of record to comply with 37 CFR 1.133. The Examiners responded favorably to the proposed amendments and arguments and said they would reconsider the rejection in view of the arguments and amendments presented herein.

1. Claims 1-30 are Patentable Over the Cited Art

The Examiner rejected claims 1-30 as obvious (35 U.S.C. §103(a)) over Choy (U.S. Patent No. 5,551,027), Cornwell (U.S. Pub. No. 2002/0032678), and SGI (Linux Man page for fetch). Applicants traverse these rejections for the following reasons.

Amended claims 1, 13, and 19 concern accessing data in a database table, and require: receiving a fetch request to fetch data from a base table that satisfies a query predicate, wherein rows of the base table are stored in table partitions and wherein there is one index partition for each determined table partition, wherein each index partition includes nodes, wherein each node in each index partition includes at least one key column value from a corresponding table row in the table partition associated with the index partition and a location identifier identifying the corresponding table row in the corresponding table partition; determining a set of nodes from a plurality of index partitions whose key column value satisfies the query predicate; ordering the set of determined nodes from the index partitions; selecting one node from the ordered set based on a position of the node in the ordering; and returning data from the table row identified by the location identifier in the selected node in response to the fetch request.

Applicants amended these claims to recite that determining the set of nodes comprises determining nodes from a plurality of index partitions, sorting the set of determined nodes from the index partitions, and selecting one node from the ordered set based on a position of the node in the ordering. The added requirements are disclosed on at least para. [0028], pgs. 11-12 and FIG. 5 of the Specification.

During the phone interview, the Examiner and the Supervisory Examiner indicated that the above discussed amendments appear to distinguish over the cited art and said they would reconsider the rejections if such amendments were submitted made in an RCE. Applicants

submit that the amended claims discussed during the phone interview, which are submitted herein, are patentable over the cited art for the following reasons.

The Examiner cited col. 11, lines 24-27 of Choy with respect to the claim requirement of determining a set of nodes, which as amended recites determining a set of nodes from a plurality of index partitions whose key column value satisfies the query predicate. (Final Office Action, pg. 3) Applicants traverse with respect to the amended claims.

Choy discusses a unique identifier of a partition, referred to as a "PID". (Choy, col. 7, lines 19-21). Choy discusses a two-tiered indexing method that creates a Local Index Table for each partition of the database and creates a Coarse Global Index Table containing one unique global index entry for each distinct local index key value in each local index table. The local index table has one local index entry for each object of interest in the corresponding partition of the table. (Choy, col. 8, lines 42-54) The global index table has the key values and for each key value the PID or local index having that key value. (Choy, col. 10, lines 25-31)

The cited col. 11 mentions that if there is a global index, qualified PIDs, i.e., PIDs of key values satisfying the query predicate, are obtained from that global index. The PIDs are sorted, duplicates are removed and the PIDs are merged with the PIDs based on the partition key. The query is then sent to each identified partition (PID). (Choy, col. 11, lines 37-40) The global index is used to direct the access request to the target partitions for processing. (Choy, col. 5, lines 19-42)

Nowhere does the cited col. 11 anywhere teach or suggest determining a set of nodes from a plurality of index partitions and then sorting the set of determined nodes. Instead, the cited Choy discusses processing a global index table to determine partitions (PIDs) having key values satisfying the query and then sending the queries to the local indexes to handle. Applicants submit that the cited Choy teaches away from the claim requirement of determining nodes from the indexes because Choy determines key values satisfying the query predicate from the global index, not index partitions as claimed.

Further, the cited Choy does not teach or suggest ordering the set of nodes determined from multiple index partitions. Instead, the cited Choy discusses obtaining PIDs from the coarse index, sorting the PIDs, removing duplicates and merging with PIDs based on the partition key. Nowhere does the cited Choy anywhere teach or suggest determining nodes from the index

partitions whose key column value satisfies the query predicate and then ordering the nodes determined from the index partitions.

The Examiner cited col. 11, lines 43-47 of Choy as teaching the claim requirement of selecting one node requirement, which now recites selecting one node from the ordered set based on a position of the node in the ordering. (Final Office Action, p. 3). Applicants traverse with respect to the amended claims.

The cited col. 11 discusses how each node having a local index may use a different query evaluation plan to evaluate the query. Nowhere does this disclose selecting one node from an ordered set, where the ordered set comprises a set of nodes comprising entries in the index partitions that satisfy the query predicate. Moreover, the cited “nodes” of Choy are different than the claimed nodes because the cited nodes of Choy comprise a computational node having a local index (see, Choy, col. 4, line 66 to col. 5, line 5, col. 5, line 58-62) on which a query is performed, whereas the claimed nodes comprise nodes in an index partition that have a key column value identifying a corresponding table row in the corresponding table partition.

The Examiner cited SGI and Cornwell as teaching fetch request processing and found that it would be obvious to modify Choy’s global index and local index processing scheme to be applied to fetch request processing. (Final Office Action, pgs. 3-4) Applicants submit that even if one made this modification, this proposed modification still fails to teach or suggest the above discussed deficiencies of Choy in that the claimed technique for processing index partition to determine nodes that satisfy the query predicates is different from the cited scheme of Choy using a global index and local indexes.

Moreover, Applicants submit that there is still no teaching of using the claimed index partitions to determine qualifying nodes for a fetch request as claimed. The Examiner cited col. 2, lines 55-59 of Choy as providing the motivation to modify the index partitioning scheme of Choy to be applied to processing fetch requests. (Final Office Action, pg. 4) Applicants traverse.

The cited col. 2 mentions that indexes are maintained on a search field to provide search efficiency. Applicants submit that although indexes provide search efficiency, there is no teaching or suggestion here that the particular described index scheme of Choy be used for fetch requests. For instance, there is no suggestion or motivation to use with search requests the claimed indexing scheme of one index partition for each determined table partition, wherein each index partition includes nodes, wherein each node in each index partition includes at least one

key column value from a corresponding table row in the table partition associated with the index partition and a location identifier identifying the corresponding table row in the corresponding table partition.

Accordingly, claims 1, 13, and 19 are patentable over the cited art because the cited art does not teach or suggest the combination of claim requirements.

Claims 2-12, 14-18, and 20-30 are patentable over the cited art because they depend from claims 1, 13, and 19, which are patentable over the cited art for the reasons discussed above. Moreover, the below discussed dependent claims provide additional ground of patentability over the cited art.

Claims 2, 14, and 20 depend from claims 1, 13, and 19 and further require: determining whether to modify a direction of the fetch request, wherein the direction comprises the direction in which the index partitions are scanned to determine nodes whose key column values satisfy the query predicate; modifying the direction of the fetch request if the determination is made to modify the fetch request; and determining the set of nodes based on the direction of the fetch request.

The Examiner cited para. [0093] of Cromwell as teaching the claim requirement of determining whether to modify a direction of a fetch request and cited para. [0099] as teaching modifying the direction of the fetch request. (Final Office Action, pg. 4) Applicants traverse.

The cited paras. [0093] and [0099] discuss that cursors can fetch forward and para. [0094] mentions that the database program may perform a FETCH backwards operations to fetch backwards.

Although the cite paragraphs of Cromwell discuss fetching in a forward and backward direction, nowhere do the cited paragraphs anywhere teach or suggestion modifying a direction of the fetch request, which is the direction in which the index partitions are scanned. Instead, the cited paragraphs discuss fetching in one direction, but do not teach or suggest determining whether to modify a direction of the fetch request and then modifying that direction.

Further, nowhere do the cited paragraphs teach or suggest determining whether to modify the direction in which the index partitions are scanned.

Accordingly, amended claims 2, 14, and 20 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 3 and 21 depend from claims 2 and 20 and further recite determining whether to modify the direction of the fetch request is based on a current fetch direction and whether the current fetch direction is opposite an ordering of the index partitions.

The Examiner cited para. [0099] of Cornwell as teaching the additional requirements of these claims. (Final Office Action, pg. 5). Applicants traverse.

The cited para. [0099] mentions performing a FETCH ABSOLUTE of k, where k is the number of rows to fetch forward for a +k or negative (-k). The data manager determines the absolute row number of the entry in the result table pointed to by the cursor and the page including the entry, and determines the relative distance of the requested entry from the current entry as the absolute value of K.

Although the cited para. [0099] discusses how to determine the relative distance of a current result table entry to the entry to which the fetch command wants to fetch, the cited paragraph nowhere teaches or suggests the claim requirement of determining whether to modify the direction of a fetch request, wherein the direction indicates the direction in which the index partition is scanned. Instead, the cited para. [0099] discusses determining a relative distance to fetch. There is no teaching or suggestion of determining whether to modify the direction of a fetch request based on whether a current fetch direction is opposite an ordering of the index partitions.

The Examiner argues that with a fetch absolute, the operation can fetch forward or backward depending on the current direction. (Final Office Action, pgs. 12-3) Applicants traverse this finding.

The cited fetch absolute determines which direction to fetch, but nowhere teaches or suggests changing or modifying the direction of the fetch based on a current fetch direction and whether the current fetch direction is opposite an ordering of the index partition. Instead, the cited fetch absolute determines a relative distance of the desired row from the current row and fetches toward that entry.

The Examiner has not cited any part of Cromwell that teaches or suggests modifying a direction of the fetch request. Instead, the cited Cromwell discusses determining the direction to go in, not changing the direction. Further, the Examiner has not cited any part of Cromwell that teaches or suggests determining whether to modify the fetch direction based on a current fetch direction and whether the current fetch direction is opposite an ordering of the index partition

Accordingly, claims 3 and 21 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 4, 15, and 22 depend from claims 2, 14, and 20 and further require that modifying the direction of the fetch request comprises: setting the fetch direction to backward if the fetch direction is backward and the fetch direction is not opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is opposite the ordering of the index partitions; and setting the fetch direction to forward if the fetch direction is backward and the fetch direction is opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is not opposite the ordering of the index partitions.

The Examiner cited para. [0099] of Cornwell as teaching the additional requirements of these claims. (Final Office Action, pgs. 4-5) Applicants traverse for the following reasons.

The cited para. [0099] discusses a fetch absolute operation, which is the number of rows to fetch forward or backward from the first entry in the result table. This is performed by determining the distance from the current entry to the requested entry, and then convert this command to a fetch relative to fetch to the requested position.

Nowhere does the cited para. [0099] anywhere teach or suggest modifying the direction of the fetch request based on the partition index ordering. Instead, the cited para. [0099] discusses how to fetch to an absolute requested row from the current position, not to change the direction of a fetch based on the index ordering. Further, there is no teaching or suggestion in the cited para. [0099] of considering an index ordering when determining how to access a row at an absolute position in a result table.

As discussed, the Examiner argues that a fetch absolute can fetch forward or backward. (Final Office Action, pg. 13) However, the fetch absolute does not change the direction of a fetch request. Instead, the FETCH ABSOLUTE command is a request to fetch k from the current row in the result table. The data manager must then determine the relative distance from the row to fetch to and the current row, and how to access that requested row. There is no teaching or suggestion of modifying the direction of the fetch based on whether the fetch direction is opposite the ordering of the index partitions.

Further, the Examiner has not cited any part of Cromwell and paragraphs of setting the fetch direction to backward if the fetch direction is backward and the fetch direction is not

opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is opposite the ordering of the index partitions and setting the fetch direction to forward if the fetch direction is backward and the fetch direction is opposite the ordering of the index partitions or if the fetch direction is forward and the fetch direction is not opposite the ordering of the index partitions. These specific operations to set the direction based on the fetch direction and the ordering of the index partitions is nowhere taught or suggested in the cited art.

Accordingly, claims 4, 15, and 22 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 5 and 23 depend from claims 2 and 20 and further require that if the fetch request is a first fetch of the fetch request, then selecting one node starting from one of: a lowest key value from each index partition if the fetch direction is forward or highest key value from each index partition if the fetch direction is backward.

The Examiner cited para. [0087] as teaching the additional requirements of these claims. (Final Office Action, pg. 5)

The cited para. [0087] discusses positioning the cursor to the position specified in the fetch, e.g., prior, first, last, current, etc if the fetch is insensitive and then returning the row positioned at the cursor. If the returned row was previously fetched, with a fetch sensitive it would reflect any changes made to the base table prior to the fetch sensitive operation.

Although the cited para. [0087] discusses how to position the cursor on a row depending on the fetch specified, there is no teaching or suggestion of selecting a node from a lowest or highest key value from each index partition depending on the fetch direction. There is no mention in the cited para. [0087] of selecting key values from index partitions as claimed. Instead, the cited para. [0087] discusses how to fetch to a requested position and then return the data from the row to which the cursor points, which may comprise changed data if the previous fetch was fetch sensitive.

Accordingly, claims 5 and 23 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 6-10, 16, and 24-29 provide further limitations concerning the direction of the fetch request and index partitions. The Examiner cited other sections of Cromwell as teaching

the additional requirements of these claims that concern how to fetch in different directions in a result table to retrieve a row from a result table. Nowhere does the cited Cromwell teach or suggest the additional requirements of these claims providing additional requirements concerning index partitions on table partitions. Instead, the cited Cromwell discusses how to fetch forward or backward through a result table whose sequential rows are on multiple pages.

Claims 11, 17, and 29 depend from claims 1, 13, and 19 and further require discarding the cached keys if the fetch request is in an opposite direction of a previous fetch request; determining a new set of nodes from each index partition; and caching the determined new set of nodes when performing the fetch operation.

The Examiner cited para. [0097] of Cromwell as teaching discarding the cached keys if the fetch request is in an opposite direction of a previous fetch request. (Final Office Action, pgs. 7, 14) Applicants traverse.

The cited para. [0097] discusses how the data manger fetches a number of pages from storage. The database program uses a statistical consideration to determine whether rows are being sequentially accessed in order to determine whether to prefetch improve performance. Although the cited para. [0097] discusses using a statistical algorithm to determine whether to prefetch for sequential access, the Examiner has not cited any part of para. [0097] that teaches or suggests discarding cached keys if the fetch request is in an opposite direction of a previous fetch request. The cited para. [0097] discusses detecting sequential access. However, there is no mention or teaching of the claim requirement of discarding cached keys if the fetch request is in an opposite direction of the previous fetch request.

Accordingly, claims 11, 17, and 29 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Applicants amended claims 12, 18, and 30 to depend from claims 11, 17, and 29.

2. Added Claims 31-36 are Patentable Over the Cited Art

Added claims 31, 33, and 35 depend from claims 1, 13, and 19 and further require determining whether the key value of the selected node from the ordered set satisfies the query predicate and selecting a next node form the ordered set following the selected node that odes not satisfy the query predicate.

The additional requirements of these claims are disclosed on at least para. [0028], pgs. 11-12 and FIG. 5 of the Specification.

Applicants submit that claims 31, 33, and 35 are patentable over the cited art because they depend from claims 1, 13, and 19 and because the additional requirements of these claims in combination with the base claims provide further grounds of patentability over the cited art.

Added claims 32, 34, and 36 depend from claims 1, 13, and 19 and further require that determining the set of nodes from the index partitions comprises executing parallel tasks to process the index partitions.

The additional requirements of these claims are disclosed on at least para. [0018], pg. 7 and para. [0047], pg. 19.

Applicants submit that claims 32, 34, and 36 are patentable over the cited art because they depend from claims 1, 13, and 19 and because the additional requirements of these claims in combination with the base claims provide further grounds of patentability over the cited art.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1-36 are patentable over the art of record. Applicants submit herewith the fee for the added claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: November 6, 2006

By: /David Victor/

David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984